The eDG
Developer Advisory

“Deploying Versata Applications to the ITS Hosting Environment”

Issued Date: .... July 30, 2001
Submitted by: .... **Joe Leary**

Purpose:
In order to deploy an HTML based application developed in Versata to the ITS hosting environment, ITS requires several files. This document is provided to assist ITS customers in creating the necessary files for ITS.

Instructions:

> **Note:** *For the purposes of this document, it is assumed that the local development environment consists of a Windows NT/2000 machine with the Versata Studio software installed. By default, the software used by Versata Studio is IBM HTTP Server (branded Apache), DB2, and IBM WebSphere. The default directory where the Studio software is installed will be referred to as “*`<Versata5.1_EJB>`*.” Any reference to the repository and application will be displayed as “*`<repository>`*” and “*`<application>`*.” The default directory for the HTTP server will be referred to as “*`<HTTP Server>`*.” These four references will have to be replaced with the local drive and directory structure implemented on the client workstation.*
>
> **Note:** *ITS will assume that Oracle is the database used for storing the business objects. Within the hosting environment, Oracle JDBC drivers are installed for accessing the ITS mainframe instance of Oracle. Any other database will require that the agency provide ITS with the necessary JDBC driver and instructions for connecting to the alternate database.*
>
> **Note:** *All directory and file names are case sensitive. It is imperative that long file names be supported throughout this process or the application will not deploy successfully. Close attention needs to be paid to the case sensitivity of all the files and directories.*
>
> **Note:** *ITS may elect to set up and deploy each application in a separate instance of the VLS. In other words, one VLS for each application These instructions assume this is the case.*

Continued …

__Deploying the business objects__

1. In Versata Studio, open the repository where the application is located. Open the HTML application. Open the “Deployment Manager.” Select “Versata Logic Server” and deploy the business objects to the local development environment. This process will create the files required to deploy the objects to the ITS hosting environment.

2. A repository .jar file containing all the Java class files will be created as `<Versata5.1_EJB>\VlsComponents\classes\<repository>.jar`.

3. A repository deployment properties file will be created as `<Versata5.1_EJB>\Vls\bin\vlsdeploy.properties`.

4. Numerous .dat files and .VJDeploy files will be created in `<Versata5.1_EJB\VlsComponents\Admin`.

5. To facilitate the transfer of these files, it is recommended that three directories be set up on the local PC. These should be called obj1, obj2, and obj3.

   - Copy the .jar files referred to in step (2) into the obj1 directory.

   - Copy the properties file referred to in step (3) into the obj2 directory.

   - Copy the remaining files referred to in step (4) into the obj3 directory.

   - Using a tool such as Winzip, zip all these directories and files into one archive file called obj.zip and then email this file to the **ITS Web Hosting Group**.

Continued …

Deploying the HTML Application

1. In Versata Studio, open the repository where the application is located. Open the HTML application. Open the “Deployment Manager.” Select “Client Application.” and deploy the HTML application to the local development environment. Make sure to check all the Recursive boxes for files, Images, and the login files during the wizard execution. This process will create the files required to deploy the user interface, servlet class, server pages, and the application logic to the ITS hosting environment.

2. To facilitate the transfer of these files, it is recommended that four directories be set up on the local PC. These should be called app1, app2, app3, and app4. The following steps describes what files should be copied into these directories.

3. Copy the application servlet.class file from <Versata5.1_EJB>\AppServer\servlets\<repository>\<application>\<application>Servlet.class to the app1 directory.

4. Copy the application .class and .html and image files, including any sub-directories, from <Versata5.1_EJB>\VlsComponents\classes\<repository>\<application>*.* to the app2 directory.

5. Copy the application deployment file from <Versata5.1_EJB>\Vls\bin\appdeploy.properties to the app3 directory.

6. Copy the HTTP files, including any sub-directories from <HTTP Server>\htdocs\<repository>\<application>*.* to the app4 directory.

7. Using a file compression tool such as Winzip, zip all these directories and files into an archive file named “app.zip,” then email the file to the **ITS Web Hosting Group**.

Continued …

ITS Deployment (Internal)

Note: *The following steps might result in existing files being overwritten. This is a normal process.*

Note: *For purposes of this deployment, the* `versata_test` *instance of the VLS will be used. This name can be replaced as necessary.*

1. Copy the .jar file from the obj1 directory to `/usr/WebSphere/versata_test/VLSComponents/Classes`.

2. Edit the `vlsdeploy.properties` file in the obj2 directory. Change the AppServer line to reflect the specific instance of the VLS you want to deploy the application to, such as "`versata_test`". Copy the `vlsdeploy.properties file` from the obj2 directory to `/usr/WebSphere/versata_test/VLS/`bin

3. Copy all the files and sub-directories from the obj3 directory to `/usr/WebSphere/versata_test/VLSComponents/Admin`

4. Run the deployment shell script to deploy the business objects to the VLS by executing `/usr/WebSphere/versata_test/VLS/bin/vlsdeploy.sh`. When asked for the Versata install directory, enter "`/usr/WebSphere/versata_test`."

5. Bring up the VLS Console by running `/usr/WebSphere/versata_test/VLS/bin/VLSConsoleEJB.sh` and verify that the business objects are deployed. It may be necessary to assign the objects to their own data server, configure the server to access the database, and modify user security to allow access to the new business objects.

6. Create two new directories for the repository and application in `/usr/WebSphere/AppServer/servlets/<repository>/<application>` and copy the file from the app1 directory into this new application directory.

Continued …

7. Create two new directories for the repository and application in `/usr/WebSphere/versata_test/VLSComponents/Classes/<repository>/<application>` and copy the files and any sub-directories from the app2 directory into this new application directory.

8. Edit the `appdeploy.properties` file in the app3 directory. Change the AppServer line to reflect the specific instance of the VLS you want to deploy the application to, such as "`versata_test`". Copy the file from the app3 directory to `/usr/WebSphere/versata_test/VLS/bin`.

9. Create two new directories for the repository and application in `/usr/netscape/suitespot/docs/<repository>/<application>` and copy the files and any sub-directories from the app4 directory into this new application directory. It is imperative that these new directories and files have read and execute rights. Use "`chmod –R 777 <repository>`."

10. Run the WebSphere console by executing `/usr/WebSphere/AppServer/bin/adminclient.sh` and stop the Versata application.

11. Run the application deployment script by executing /`usr/WebSphere/versata_test/VLS/bin/appdeploy.sh` to deploy the application. When asked for the "htdocs" root directory, enter "`/usr/netscape/suitespot/docs`."

12. Start the Versata application in the WebSphere console and verify the application has deployed properly. It may be necessary to refresh the screen to see the new application.

13. Test the application by bringing up a web browser and going to `http://<system>/webapp/<repository>/<application>`.